

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук
(повна назва)

Кафедра _____ Інформаційних управляючих систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____ другий (магістерський)

_____ Дослідження методів оцінювання складності
_____ проєктів розробки ігор віртуальної реальності
_____ (тема)

Виконав:
здобувач _____ 2 _____ року навчання,
групи _____ УПГІТм-24-1
_____ Владислав ОБЕРШЕВ
(Власне ім'я, ПРІЗВИЩЕ)

Спеціальність _____ 122 – Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми _____ освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Управління проектами в
галузі інформаційних технологій
(повна назва освітньої програми)

Керівник _____ проф. каф. ІУС Сергій ЧАЛИЙ
(посада, Власне ім'я, ПРІЗВИЩЕ)

Допускається до захисту
Зав. кафедри

_____ Константин ПЕТРОВ
(підпис) (Власне ім'я, ПРІЗВИЩЕ)

2026 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
 Кафедра _____ Інформаційних управляючих систем _____
 Рівень вищої освіти _____ другий (магістерський) _____
 Спеціальність _____ 122 Комп'ютерні науки _____
 (код і повна назва)
 Тип програми _____ освітньо-наукова _____
 (освітньо-професійна або освітньо-наукова)
 Освітня програма _____ Управління проектами в галузі інформаційних технологій _____
 (повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
 (підпис)

« 06 » квітня 20 26 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Обершеву Владиславу Олександровичу _____
 (прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження методів оцінювання складності проєктів розробки ігор віртуальної реальності»

затверджена наказом університету від 23 березня 2026 р. № 231Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 20 травня 2026 р.

3. Вихідні дані до роботи особливості процесів управління проектами розробки VR-ігор; метрики візуального програмування Blueprint в Unreal Engine; технології Unreal Engine 5, Blueprint System, Python API, Jira REST API; методи параметричного моделювання, статистичного аналізу та оцінювання точності прогнозування трудомісткості.

4. Перелік питань, що потрібно опрацювати в роботі аналіз сучасних підходів до управління проектами розробки VR-ігор; дослідження існуючих методів оцінювання складності IT-проєктів; розробка методу оцінювання складності проєктів на основі метрик візуального програмування Blueprint; формування системи VR-коефіцієнтів для врахування специфіки віртуальної реальності; проєктування архітектури програмного модуля системи оцінювання; реалізація алгоритмів аналізу Blueprint-графів та розрахунку метрик складності; експериментальна перевірка точності розробленого методу на реальних проєктах та аналіз ефективності впровадження методу у процеси організації.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання	06.04.2026	виконано
2	Аналіз предметної галузі і постановка задачі	07.04.2026	виконано
3	Розробка удосконаленого методу оцінювання складності проєктів ігор віртуальної реальності	13.04.2026	виконано
4	Розробка проєкту впровадження методу оцінювання складності в студії розробки ігор	17.04.2026	виконано
5	Розробка програмної реалізації та експериментальна перевірка	27.04.2026	виконано
6	Підготовка пояснювальної записки	04.05.2026	виконано
7	Підготовка презентації та доповіді	10.05.2026	виконано
8	Перевірка на плагіат	13.05.2026	виконано
9	Нормоконтроль	14.05.2026	виконано
10	Рецензування	15.05.2026	виконано
11	Попередній захист	16.05.2026	виконано
12	Занесення диплома в електронний архів	17.05.2026	виконано
13	Допуск до захисту у зав. кафедри	18.05.2026	виконано

Дата видачі завдання 06 квітня 2026 р.

Здобувач _____
(підпис)

Керівник роботи _____ проф. каф. ІУС Сергій Чалий
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка містить: 70 с., 2 рис., 8 табл., 2 дод. 30 джерел.

ВІЗУАЛЬНЕ ПРОГРАМУВАННЯ, ВІРТУАЛЬНА РЕАЛЬНІСТЬ, ОЦІНЮВАННЯ СКЛАДНОСТІ, ТРУДОМІСТКІСТЬ ПРОЄКТУ, УПРАВЛІННЯ ІТ-ПРОЄКТАМИ, BLUEPRINT, UNREAL ENGINE.

Об'єктом дослідження є процеси управління проєктами розробки ігор віртуальної реальності.

Метою роботи є підвищення точності оцінювання складності проєктів розробки ігор віртуальної реальності шляхом розробки методу, що враховує метрики візуального програмування та специфічні характеристики VR-систем.

Методи розробки та проєктування включають аналіз існуючих моделей оцінювання, застосування метрик теорії графів для оцінювання Blueprint-логіки, введення спеціалізованих коефіцієнтів складності VR, методи параметричного моделювання, програмну реалізацію на основі Unreal Engine та Python, а також статистичні методи оцінювання точності MAPE та кореляційний аналіз.

У результаті кваліфікаційної роботи було розроблено метод оцінювання складності VR-проєктів, що поєднує параметричні моделі з метриками візуального програмування, створенні програмного інструменту для автоматизації оцінювання, а також експериментальній перевірці методу на реальних проєктах.

ABSTRACT

The explanatory note contains: 70 pages, 2 figures, 8 tables, 2 appendices, 30 sources.

BLUEPRINT, PROJECT EFFORT ESTIMATION, PROJECT MANAGEMENT, SOFTWARE COMPLEXITY, UNREAL ENGINE, VIRTUAL REALITY, VISUAL PROGRAMMING.

The object of research is the processes of managing virtual reality game development projects.

The aim of the work is to increase the accuracy of assessing the complexity of virtual reality game development projects by developing a method that takes into account visual programming metrics and specific characteristics of VR systems.

Development and design methods include analysis of existing evaluation models, application of graph theory metrics for Blueprint logic evaluation, introduction of specialized VR complexity factors, parametric modeling methods, software implementation based on Unreal Engine and Python, as well as statistical methods for assessing MAPE accuracy and correlation analysis.

As a result of the qualification work, a method for assessing the complexity of VR projects was developed, which combines parametric models with visual programming metrics, the creation of a software tool for automating the assessment, as well as experimental verification of the method on real projects.

ЗМІСТ

С.

Скорочення та умовні позначки	8
Вступ.....	9
1 Аналіз предметної області та постановка задачі	10
1.1 Аналіз процесів управління проектами розробки ігор віртуальної реальності.....	10
1.2 Аналіз методів оцінювання складності ІТ-проектів.....	14
1.3 Аналіз методів оцінювання складності ІТ-проектів на ігрових двигунах	19
1.4 Постановка задачі дослідження.....	23
2 Удосконалений метод оцінювання складності проектів ігор віртуальної реальності.....	25
2.1 Підхід до оцінювання складності проектів розробки ігор віртуальної реальності.....	25
2.2 Розроблений метод оцінювання трудомісткості проектів віртуальної реальності.....	30
3 Розробка проекту впровадження методу оцінювання складності в студії розробки ігор	34
3.1 Ініціалізація та розробка концепції проекту впровадження.....	34
3.2 Розробка життєвого циклу проекту впровадження	36
3.3 Структуризація проекту впровадження.....	37
3.4 Проектування системи оцінювання складності проектів віртуальної реальності.....	40
3.5 Керування часом, вартістю та ресурсами проекту	44
4 Програмна реалізація та експериментальна перевірка	47
4.1 Реалізація системи автоматизованого оцінювання складності проектів	47
4.2 Експериментальна перевірка методу на реальних проектах	52

4.3 Оцінка впливу впровадження на процеси організації.....	54
Висновок	57
Перелік джерел посилання	58
Додаток А. Лістинг програми	61
Додаток Б. Графічний матеріал кваліфікаційної роботи	63

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

IT – інформаційні технології

API – application programming interface

BI – Business Intelligence

CI/CD – continuous integration/continuous deployment

COCOMO – constructive cost model

FPS – frame per seconds

FTE – full-time equivalent

GDCM – game development complexity model

JSON – JavaScript object notation

MAPE – mean absolute percentage error

PCVR – personal computer virtual reality

PERT – program evaluation and review technique

PM – project manager

VR – virtual reality

VICI – virtual reality interaction complexity index

VPM – virtual reality productivity multiplier

WBS – work breakdown structure

ВСТУП

Розробка ігор віртуальної реальності є складним процесом, що поєднує програмну інженерію, комп'ютерну графіку, геймдизайн та управління проєктами. На відміну від традиційних програмних систем, virtual reality (VR) додатки мають підвищені вимоги до продуктивності, інтерактивності та якості користувацького досвіду, що ускладнює їх розробку та планування.

Однією з ключових проблем у цій сфері є оцінювання складності та трудомісткості проєктів. Існуючі методи оцінювання, розроблені переважно для класичних програмних систем, лише частково враховують специфіку VR. Зокрема, вони не враховують особливості візуального програмування, вимоги до стереоскопічного рендерингу та складність інтерактивної взаємодії. Унаслідок цього виникає недооцінювання термінів виконання робіт і перевищення бюджету проєктів.

Актуальність теми зумовлена необхідністю створення спеціалізованих методів оцінювання, які враховують особливості сучасних ігрових рушіїв і технологій віртуальної реальності. Важливу роль у цьому відіграють системи візуального програмування, що потребують нових підходів до оцінювання складності на основі метрик теорії графів.

Метою роботи є розробка методу оцінювання складності проєктів VR-ігор для підвищення точності прогнозування трудомісткості шляхом поєднання параметричних моделей, метрик візуального програмування та спеціалізованих коефіцієнтів.

Практична значущість роботи полягає у можливості використання розробленого методу в невеликих студіях розробки ігор для підвищення точності планування, зниження ризиків і оптимізації використання ресурсів.

При підготовці документації були дотримані чинні державні стандарти України [1-2].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз процесів управління проєктами розробки ігор віртуальної реальності

Розробка ігор віртуальної реальності є складним багатокomпонентним процесом, що поєднує програмну інженерію, комп'ютерну графіку, геймдизайн та управління проєктами. На відміну від традиційної розробки програмного забезпечення, VR-проєкти характеризуються підвищеними вимогами до продуктивності, інтерактивності та комфорту користувача, що безпосередньо впливає на структуру процесів управління такими проєктами.

Життєвий цикл проєкту розробки гри віртуальної реальності зазвичай включає декілька основних етапів: формування концепції, прототипування, виробництво, тестування користувацького досвіду та реліз програмного продукту. На етапі концепції формується загальна ідея гри, визначається цільова аудиторія та платформа віртуальної реальності, а також розробляється концепт-документ, що описує основні механіки, художній стиль та технічні вимоги.

Етап прототипування спрямований на створення експериментальної версії гри або окремих ігрових механік з метою перевірки їх працездатності у VR-середовищі. У цьому процесі широко використовуються можливості сучасних ігрових рушіїв, зокрема системи візуального програмування Blueprint в Unreal Engine, що дозволяє швидко реалізовувати прототипи ігрової логіки без значного обсягу текстового програмного коду [3].

Основна фаза розробки включає реалізацію ігрових механік, створення тривимірних моделей, анімацій, інтеграцію звукового супроводу та оптимізацію продуктивності. Особливістю VR-проєктів є необхідність підтримання стабільної частоти кадрів на рівні не менше 90 Frames per seconds (FPS), оскільки нижчі значення можуть призводити до виникнення дискомфорту або симуляторної хвороби у користувачів. Саме тому оптимізація графічного рендерингу та мінімізація затримки

між рухом користувача та відображенням у VR-шоломі є ключовими технічними задачами.

Управління розробкою VR-проектів також передбачає організацію ітеративного процесу планування робіт. У багатьох студіях розробки використовується підхід Agile, що реалізується через короткі ітерації, протягом яких команда створює функціональні частини гри та регулярно перевіряє їх на тестових пристроях.

Вхідними даними процесу управління VR-проектом виступають концепт-документ, технічне завдання, вимоги до платформи віртуальної реальності а також обмеження бюджету та ресурсів команди. У процесі виконання проекту формуються проміжні результати у вигляді тестових збірок програмного продукту, технічної документації та метрик продуктивності системи. Кінцевим результатом є готовий VR-застосунок, що може бути опублікований на відповідних цифрових платформах розповсюдження.

Важливою особливістю управління VR-проектами є наявність технічних обмежень, пов'язаних з апаратними характеристиками VR-пристроїв. Наприклад, автономні VR-шоломи мають обмежені обчислювальні ресурси, що змушує розробників оптимізувати кількість полігонів, текстур та графічних ефектів. Крім того, більшість незалежних VR-проектів реалізуються командами розробників чисельністю від трьох до десяти осіб, що додатково ускладнює планування та управління проектом.

Проекти розробки ігор віртуальної реальності можуть суттєво відрізнятися за масштабом, технологічними характеристиками та типами взаємодії користувача з віртуальним середовищем. Для ефективного управління такими проектами доцільно застосовувати їх класифікацію за кількома ключовими ознаками.

Першою важливою ознакою є масштаб проекту та тривалість його реалізації. За цим критерієм можна виділити короткі VR-досвіди, середні VR-ігри та великі VR-проекти. Короткі VR-досвіди зазвичай розробляються протягом 2-6 місяців невеликими командами та мають обмежений обсяг контенту. Середні VR-

ігри можуть розроблятися від шести до вісімнадцяти місяців та включають повноцінний ігровий процес з декількома рівнями або сценаріями. Реалізація найбільш масштабних VR-проектів може тривати понад півтора року та потребує значних фінансових і людських ресурсів [4].

Другою важливою ознакою класифікації є тип взаємодії користувача з віртуальним середовищем. У сучасних VR-іграх широко використовуються різні механіки взаємодії, серед яких можна виділити VR-симулятори, VR-екшн-ігри та VR-квести. VR-симулятори орієнтовані на відтворення реалістичних фізичних процесів або середовищ, наприклад тренажерів або симуляторів транспорту. VR-екшн-ігри характеризуються швидкою динамічною взаємодією з об'єктами та противниками у віртуальному просторі. Натомість VR-квести мають більш виражений наративний компонент та зосереджені на розв'язанні логічних задач і дослідженні віртуального середовища [5].

Ще одним критерієм класифікації є платформа віртуальної реальності, для якої розробляється програмний продукт. Серед найбільш поширених платформ можна виділити автономні VR-пристрої, персональні VR-системи та кросплатформні рішення. Автономні VR-пристрої мають обмежені обчислювальні ресурси, але забезпечують високу мобільність та доступність для користувачів. VR-системи на персональних комп'ютерах використовують обчислювальні ресурси, що дозволяє реалізувати більш складну графіку та фізичні симуляції.

Окрім того, VR-проекти можна класифікувати за архітектурними особливостями програмної реалізації. У сучасних ігрових рушіях, таких як Unreal Engine, широко застосовується система візуального програмування Blueprint, що дозволяє реалізовувати ігрову логіку у вигляді графів. У деяких проєктах використовується переважно Blueprint-архітектура, тоді як інші проєкти комбінують візуальне програмування з використанням мови програмування C++ для оптимізації продуктивності та реалізації складних алгоритмів [6].

Однією з ключових проблем управління проектами розробки ігор віртуальної реальності є оцінювання їхньої складності та прогнозування трудомісткості виконання робіт. У традиційній розробці програмного забезпечення широко застосовуються різноманітні методи оцінювання, однак більшість із них не враховує специфічні особливості VR-систем.

Першою суттєвою проблемою є відсутність стандартизованих метрик для оцінювання складності ігрових механік віртуальної реальності. Багато ключових елементів VR-досвіду, таких як системи телепортації, просторової навігації або взаємодії з об'єктами, мають значно складнішу реалізацію порівняно з традиційними інтерфейсами користувача. Проте у більшості існуючих моделей оцінювання складності програмних систем ці фактори не враховуються.

Другою проблемою є недооцінка трудомісткості VR-специфічних технічних завдань. Зокрема, реалізація стереоскопічного рендерингу потребує подвійного обчислення зображення для кожного ока користувача, що значно збільшує навантаження на графічний процесор. Крім того, розробникам необхідно проводити регулярне тестування користувацького досвіду для запобігання виникненню симуляторної хвороби, що також потребує додаткових ресурсів і часу.

Ще однією проблемою є ризик накопичення технічного боргу у системах візуального програмування. У процесі швидкого прототипування розробники часто створюють складні графи логіки у системі Blueprint, які з часом стають важкими для підтримки та модифікації. Відсутність чітких метрик складності таких графів ускладнює прогнозування трудомісткості їх подальшої оптимізації або рефакторингу.

Крім того, процес розробки VR-ігор характеризується високим рівнем експериментальності. Багато ігрових механік перевіряються безпосередньо під час розробки, оскільки їх ефективність може бути оцінена лише після тестування у VR-середовищі. Це призводить до значної невизначеності у плануванні строків виконання робіт та створює ризики перевищення бюджету проєкту.

Порівняння вимог до традиційного IT-проєкту та VR-проєкту буде наведено нижче у таблиці 1.1

Таблиця 1.1 – Порівняння особливостей традиційних IT-проєктів та VR-проєктів

Критерій	Традиційні IT-проєкти	VR-проєкти
Тип взаємодії	2D інтерфейс	3D інтерактивне середовище
Вимоги до продуктивності	Середні	Високі (90+ FPS)
Спосіб програмування	Текстовий код	Текстовий + візуальне програмування
Складність тестування	Стандартна	Підвищена через VR-пристрої
UX-фактори	Інтерфейс користувача	Просторова взаємодія

Таким чином, існуючі підходи до оцінювання складності IT-проєктів не повною мірою враховують специфіку розробки ігор віртуальної реальності. Це обумовлює необхідність розроблення нових або удосконалених методів оцінювання, які б дозволили врахувати особливості VR-механік, архітектури ігрових рушіїв та вимоги до продуктивності систем віртуальної реальності.

1.2 Аналіз методів оцінювання складності IT-проєктів

Оцінювання складності програмних проєктів є однією з ключових задач управління IT-проєктами, оскільки від точності такої оцінки залежить планування строків виконання робіт, формування бюджету, розподіл ресурсів команди та управління ризиками. У сучасній практиці розробки програмного забезпечення використовується широкий спектр методів оцінювання, які можна класифікувати на

декілька основних груп: експертні, параметричні, статистичні та методи на основі машинного навчання.

Експертні методи ґрунтуються на досвіді спеціалістів, які оцінюють складність майбутнього проєкту на основі знань про аналогічні проєкти або власного практичного досвіду. До найбільш поширених підходів цієї групи належать метод оцінки за аналогією, метод Дельфі та трьохточкова оцінка. Метод оцінки за аналогією передбачає порівняння нового проєкту з раніше реалізованими системами та використання фактичних даних щодо їх трудомісткості як основи для прогнозування. Метод Дельфі передбачає залучення групи експертів, які незалежно оцінюють складність проєкту, після чого результати узгоджуються через декілька ітерацій. Трьохточкова оцінка Program Evaluation and Review Technique (PERT) базується на визначенні оптимістичної, песимістичної та найбільш імовірної оцінки трудомісткості, що дозволяє врахувати невизначеність проєкту.

Параметричні методи оцінювання базуються на математичних моделях, які встановлюють залежність між розміром програмної системи та трудомісткістю її розробки. Найбільш відомим представником цієї групи є модель Constructive Cost Model (COCOMO) II, запропонована доктором Баррі Боуєм. У цій моделі трудомісткість проєкту визначається як функція обсягу програмного коду та набору коефіцієнтів, що відображають складність технологічного середовища, кваліфікацію команди та інші фактори [6].

Іншим поширеним підходом є метод функціональних точок, який дозволяє оцінювати розмір програмної системи незалежно від мови програмування на основі кількості функціональних компонентів системи, таких як входи, виходи, логічні файли, інтерфейси [7].

Останніми роками значного поширення набули методи прогнозування трудомісткості на основі машинного навчання. Такі підходи використовують історичні дані про попередні проєкти для побудови моделей регресії, нейронних мереж або інших алгоритмів прогнозування. Завдяки використанню великих наборів

даних такі моделі можуть виявляти складні залежності між параметрами проєкту та фактичною трудомісткістю розробки.

У галузі розробки відеоігор також використовуються спеціалізовані моделі оцінювання складності, зокрема Game development complexity model (GDCM). Ця модель враховує специфічні фактори геймдев-проєктів, такі як складність ігрової механіки, кількість ігрового контенту, складність штучного інтелекту та інтерактивності ігрового середовища.

Існуючі методи оцінювання складності програмних проєктів можна систематизувати за типом використовуваних підходів та рівнем формалізації. Експертні методи забезпечують високу гнучкість, але залежать від досвіду спеціалістів. Параметричні моделі дозволяють отримати формалізовану оцінку трудомісткості, проте потребують точного визначення параметрів системи. Методи машинного навчання забезпечують високу точність прогнозування, але потребують значних обсягів історичних даних.

Незважаючи на широке використання різних методів оцінювання складності програмних систем, їх застосування до проєктів розробки ігор віртуальної реальності має певні обмеження. Це пов'язано з тим, що VR-проєкти характеризуються специфічними технічними та дизайнерськими вимогами, які не враховуються у більшості традиційних моделей оцінювання.

Модель СОСОМО II була розроблена для оцінювання трудомісткості розробки традиційних програмних систем і базується на оцінці обсягу програмного коду. Проте у VR-проєктах значна частина логіки реалізується за допомогою систем візуального програмування, наприклад Blueprint у ігровому рушії Unreal Engine. Такі системи використовують графічні блоки замість текстового коду, що ускладнює пряме застосування метрик, заснованих на кількості рядків програмного коду.

Крім того, модель СОСОМО II не враховує специфічні технічні вимоги систем віртуальної реальності, зокрема необхідність підтримання високої частоти кадрів та мінімальної затримки відображення. Для забезпечення комфортного користувацького

досвіду у VR-системах затримка між рухом користувача та оновленням зображення повинна становити менше 20 мс, а частота кадрів має перевищувати 90 FPS. Досягнення таких показників часто потребує значних зусиль з оптимізації рендерингу, що не враховується у традиційних моделях оцінювання.

Метод функціональних точок також має обмежену застосовність до VR-проектів. Основною причиною цього є те, що цей підхід орієнтований на оцінювання інформаційних систем, де основну роль відіграють обробка даних та взаємодія з користувацькими інтерфейсами. У VR-додатках значна частина складності пов'язана з реалізацією просторової взаємодії, фізичного моделювання та тривимірної графіки, що не відображається у традиційних метриках.

Спеціалізовані моделі оцінювання складності ігрових проектів, такі як GDCM, враховують певні аспекти геймдеву, однак більшість із них розроблялися для традиційних відеоігор і не враховують специфіку VR-технологій. Наприклад, вони не враховують складність реалізації систем просторового переміщення, інтерактивних механік взаємодії з об'єктами або систем відстеження рухів рук користувача.

У практиці незалежних студій розробки ігор віртуальної реальності найбільш поширеними залишаються експертні методи оцінювання. Це пояснюється тим, що такі студії зазвичай мають обмежені ресурси та невелику кількість завершених проектів, що ускладнює застосування статистичних моделей або методів машинного навчання. У таких умовах оцінка складності проекту часто базується на досвіді технічних лідерів та попередніх проєктах команди.

Аналіз показує, що існуючі методи оцінювання складності програмних систем лише частково придатні для використання у проєктах розробки ігор віртуальної реальності. Це обумовлює необхідність розроблення спеціалізованих методів оцінювання, які враховували б специфіку VR-технологій, архітектуру сучасних ігрових рушіїв та особливості взаємодії користувача з віртуальним середовищем.

Для визначення найбільш придатних підходів до оцінювання складності проєктів розробки ігор віртуальної реальності доцільно провести порівняльний аналіз

основних методів оцінювання програмних проєктів за ключовими критеріями. Такими критеріями можуть бути точність прогнозування трудомісткості, складність застосування методу, потреба у наявності історичних даних та здатність враховувати специфіку VR-технологій.

Експертні методи характеризуються відносною простотою застосування та не потребують використання формалізованих моделей або великих обсягів статистичних даних. Водночас їх точність значною мірою залежить від досвіду експертів та наявності у них знань про аналогічні проєкти. У випадку нових або експериментальних VR-проєктів, де відсутні подібні приклади, точність таких оцінок може бути недостатньою.

Параметричні моделі, такі як COCOMO II, забезпечують більш формалізований підхід до оцінювання трудомісткості розробки. Вони дозволяють отримувати кількісні оцінки на основі об'єктивних параметрів проєкту, однак потребують точного визначення обсягу програмного коду. У VR-проєктах це може бути складним через використання систем візуального програмування та значну роль дизайнерських елементів у формуванні складності системи.

Метод функціональних точок забезпечує незалежність оцінки від конкретної технології реалізації, що є його важливою перевагою. Однак він орієнтований переважно на інформаційні системи і не враховує складність графічних компонентів, фізичного моделювання та інтерактивності, що є ключовими елементами VR-додатків.

Методи машинного навчання можуть забезпечувати високу точність прогнозування трудомісткості проєктів, якщо доступна достатня кількість історичних даних. Однак для невеликих студій розробки відеоігор, застосування таких методів може бути ускладненим через нестачу даних для навчання моделей.

Результати аналізу існуючих методів, а саме їх переваги, недоліки та придатність для VR занесено в таблицю 1.2.

Таблиця 1.2 – Методи оцінювання складності ІТ-проектів

Метод	Переваги	Недоліки	Придатність для VR
СОСОМО II	формалізований підхід	не враховує Blueprint та VR	Низька
Метод функціональних точок	незалежність від мови	не враховує 3D та VR UX	Низька
GDCM	враховує геймдев	не враховує специфіку VR	Середня
Машинне навчання	висока точність	потребує великих даних	Середня

Узагальнюючи результати аналізу, можна зробити висновок, що жоден із існуючих методів оцінювання складності програмних проектів не враховує повною мірою специфіку розробки ігор віртуальної реальності. Тому доцільним є створення удосконаленого методу оцінювання, який поєднуватиме формалізовані параметричні моделі з метриками складності систем візуального програмування та спеціальними коефіцієнтами, що враховують технологічні та дизайнерські особливості VR-проектів.

1.3 Аналіз методів оцінювання складності ІТ-проектів на ігрових двигунах

Сучасні ігрові рушії, такі як Unreal Engine, широко використовують системи візуального програмування для реалізації ігрової логіки. Одним із найпоширеніших інструментів такого типу є система Blueprint, яка дозволяє розробникам створювати поведінку ігрових об'єктів за допомогою графічних вузлів та зв'язків між ними. Такий підхід значно спрощує процес прототипування та дозволяє геймдизайнерам і

технічним художникам брати участь у програмуванні ігрових механік без використання традиційних мов програмування.

Разом з тим використання візуального програмування створює нові виклики у сфері оцінювання складності програмних систем. У класичних програмних проєктах складність системи зазвичай оцінюється за допомогою метрик, заснованих на кількості рядків коду, кількості функцій або структурі програмних модулів. Однак у системах візуального програмування такі метрики не можуть бути застосовані безпосередньо, оскільки програмна логіка представлена у вигляді графів.

Для оцінювання складності Blueprint-систем можуть використовуватися метрики теорії графів. До основних із них належать кількість вузлів, кількість зв'язків між ними та рівень вкладеності логічних конструкцій. Кількість вузлів відображає обсяг функціональних блоків, що реалізують логіку програми, тоді як кількість зв'язків характеризує складність взаємодії між різними елементами графа. Чим більшою є кількість зв'язків між вузлами, тим складнішою є структура програмної логіки.

Ще однією важливою метрикою є цикломатична складність графа виконання. Вона визначає кількість незалежних шляхів виконання програми та дозволяє оцінити складність тестування та підтримки системи. У контексті Blueprint-графів цикломатична складність визначається кількістю умовних переходів, подій та циклічних структур у графі виконання. Високі значення цієї метрики можуть свідчити про надмірну складність логіки та підвищений ризик виникнення помилок.

Окрім того, важливим показником складності системи візуального програмування є рівень залежностей між різними Blueprint-класами. У великих ігрових проєктах окремі Blueprint-класи можуть активно взаємодіяти між собою, використовуючи події, інтерфейси та системи передачі повідомлень. Надмірна кількість таких залежностей може призводити до підвищення складності підтримки системи та ускладнювати процес її модифікації.

Оцінювання складності систем візуального програмування потребує використання спеціалізованих метрик, які враховують графову структуру програмної логіки та взаємозв'язки між її компонентами.

Однією з ключових особливостей проєктів розробки ігор віртуальної реальності є складність реалізації систем взаємодії користувача з віртуальним середовищем. На відміну від традиційних відеоігор, де взаємодія відбувається через клавіатуру або геймпад, у VR-системах користувач безпосередньо взаємодіє з тривимірним простором за допомогою контролерів руху або систем відстеження рук.

Одним із основних елементів VR-архітектури є так званий VR Pawn – базовий об'єкт, що представляє користувача у віртуальному середовищі. Реалізація VR Pawn включає обробку позиції гарнітури, контролерів руху та системи відстеження положення у просторі. Якщо стандартні шаблони ігрового рушія не відповідають вимогам проєкту, розробникам доводиться створювати кастомні системи переміщення та управління камерою, що значно збільшує трудомісткість розробки.

Ще одним важливим компонентом VR-проєктів є система переміщення користувача у віртуальному просторі. Найбільш поширеними підходами є телепортація, плавне переміщення та переміщення за допомогою фізичних дій користувача. Кожен із цих підходів має різний рівень складності реалізації та різні вимоги до тестування з точки зору комфорту користувача.

Важливим аспектом VR-взаємодії є також система маніпулювання об'єктами. До базових механік належать хватання об'єктів, їх переміщення, обертання та кидання. У більш складних системах може використовуватися дворука взаємодія, що потребує синхронізації рухів обох контролерів та коректного фізичного моделювання об'єкта. Такі механіки значно збільшують складність програмної реалізації та потребують додаткового тестування.

Окрім цього, важливою складовою VR-додатків є користувацький інтерфейс. У VR-середовищах широко використовуються тривимірні віджети, які розміщуються безпосередньо у віртуальному просторі. Реалізація такого інтерфейсу потребує

врахування особливостей просторового сприйняття користувача, коректної взаємодії з контролерами руху та оптимізації продуктивності системи.

Значний вплив на складність VR-проєкту має також навантаження на систему рендерингу. У VR-додатках зображення повинно рендеритися окремо для кожного ока користувача, що фактично подвоює обсяг обчислень графічного процесора. Крім того, для забезпечення комфортного користувацького досвіду необхідно підтримувати стабільну частоту кадрів не менше 90 FPS.

Одним із ключових недоліків існуючих підходів є відсутність інтегрованих метрик, які дозволяли б одночасно враховувати складність візуального програмування та текстового коду. У сучасних ігрових рушіях значна частина логіки реалізується у вигляді Blueprint-графів, тоді як інша частина – у вигляді C++ коду. Відсутність єдиної моделі оцінювання для цих двох типів програмної логіки ускладнює прогнозування трудомісткості розробки.

Іншою важливою проблемою є нехтування фактором комфорту користувача у VR-системах. Для запобігання виникненню симуляторної хвороби розробники повинні проводити численні ітерації тестування та коригування ігрового дизайну. Такі ітерації можуть значно збільшувати тривалість розробки проєкту, однак більшість існуючих моделей оцінювання не враховує цей фактор.

Фактори складності VR-проєктів та їх впливу на оцінювання та занесемо у таблицю 1.3.

Таблиця 1.3 – Основні фактори складності VR-проєктів та проблеми їх оцінювання

Фактор складності	Прояв у VR-проєктах	Вплив на трудомісткість	Обмеження класичних методів
Візуальне програмування Blueprint	Графова структура логіки	Зростання складності підтримки та тестування	Не враховують графові метрики
Цикломатична складність Blueprint	Велика кількість умов і шляхів виконання	Ускладнення тестування та налагодження	Орієнтація лише на текстовий код

Продовження таблиці 1.3

Фактор складності	Прояв у VR-проєктах	Вплив на трудомісткість	Обмеження класичних методів
Залежності між Blueprint-класами	Події, інтерфейси	Підвищення складності модифікації	Відсутність оцінки взаємозв'язків
VR Pawn та motion control	Обробка трекінгу та контролерів	Збільшення обсягу логіки взаємодії	Не враховується специфіка VR
Системи переміщення	Teleport / Smooth locomotion	Додаткові цикли тестування	Відсутність VR UX-факторів
Взаємодія з об'єктами	Grab, throw, two-hand interaction	Підвищення складності фізики	Не оцінюється інтерактивність
VR UI	3D widgets та spatial UI	Ускладнення UX та input logic	Відсутність VR UI метрик
Вимоги до продуктивності	Render для двох очей, 90 FPS	Значні витрати на оптимізацію	<i>Не враховуються optimization cycles</i>

Таким чином, існуючі підходи до оцінювання складності програмних систем мають низку обмежень у контексті VR-проєктів. Це зумовлює необхідність розроблення спеціалізованих методів оцінювання, які враховують особливості систем візуального програмування, специфіку VR-взаємодії та вимоги до продуктивності системи.

1.4 Постановка задачі дослідження

У попередніх підрозділах було проведено аналіз процесів управління проєктами розробки ігор віртуальної реальності, а також розглянуто існуючі методи оцінювання складності IT-проєктів. Проведений аналіз показав, що сучасні методи оцінювання лише частково враховують специфіку розробки VR-додатків, що призводить до значних відхилень між плановими та фактичними показниками тривалості та вартості проєктів.

Об'єктом дослідження у даній роботі є процеси управління проєктами розробки ігор віртуальної реальності.

Предметом дослідження є методи та моделі оцінювання складності ігрових механік віртуальної реальності та трудомісткості їх реалізації на ігрових двигунах.

Метою роботи є дослідження сучасних методів оцінювання складності програмних проєктів для розроблення удосконаленого методу оцінювання складності проєктів розробки VR-ігор.

Для досягнення мети кваліфікаційної роботи необхідно вирішити такі задачі дослідження:

- проаналізувати наявні методи оцінювання складності розробки ІТ-проєктів;
- розробити модель оцінювання складності проєктів віртуальної реальності з урахуванням архітектури систем візуального програмування;
- удосконалити метод оцінювання трудомісткості проєктів віртуальної реальності шляхом введення специфічних коефіцієнтів складності;
- розробити проєкт впровадження методу оцінювання у процеси планування студії розробки ігор віртуальної реальності;
- провести експериментальну перевірку розробленого методу оцінювання складності на VR-проєктах.

Результати виконання цих задач повинні забезпечити підвищення точності прогнозування трудомісткості VR-проєктів та покращити ефективність управління процесами розробки у студіях, що спеціалізуються на створенні застосунків віртуальної реальності.

2 УДОСКОНАЛЕНИЙ МЕТОД ОЦІНЮВАННЯ СКЛАДНОСТІ ПРОЄКТІВ ІГОР ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ

2.1 Підхід до оцінювання складності проєктів розробки ігор віртуальної реальності

Проєкти розробки ігор віртуальної реальності характеризуються підвищеним рівнем складності порівняно з традиційними програмними системами або навіть звичайними відеоіграми. Це пов'язано з поєднанням кількох складних технологічних складових: інтерактивної тривимірної графіки, реального часу обробки даних, систем відстеження рухів користувача та необхідності забезпечення високої продуктивності системи. У VR-додатках навіть незначні затримки або помилки візуалізації можуть призводити до погіршення користувацького досвіду та появи так званого ефекту нудоти.

Традиційні методи оцінювання складності програмних систем, такі як метрики програмного коду або класичні моделі оцінювання трудомісткості програмного забезпечення, не повністю враховують специфіку розробки VR-систем. Зокрема, вони недостатньо відображають складність взаємодії користувача з віртуальним середовищем, складність оптимізації рендерингу та особливості використання сучасних ігрових рушіїв, таких як Unreal Engine. У зв'язку з цим виникає необхідність формування спеціалізованого підходу до оцінювання складності проєктів розробки VR-ігор. Такий підхід повинен враховувати технологічні, архітектурні, дизайнерські та організаційні фактори, що впливають на складність реалізації програмної системи.

Запропонований підхід передбачає поетапне оцінювання складності, що включає аналіз факторів складності VR-розробки, формалізацію метрик складності систем візуального програмування та визначення узагальненої послідовності оцінювання складності проєкту.

Розробка VR-додатків відрізняється від традиційної розробки програмного забезпечення значною кількістю специфічних факторів, які впливають на складність реалізації проєкту. До таких факторів належать технологічні, архітектурні, дизайнерські та проєктні аспекти.

Технологічні фактори визначаються особливостями роботи апаратного та програмного забезпечення віртуальної реальності. Однією з ключових характеристик VR є використання стереоскопічного рендерингу, при якому сцена відображається окремо для кожного ока користувача. Це фактично подвоює навантаження на графічний процесор, оскільки система повинна формувати два зображення з різних точок огляду.

Ще одним важливим технологічним фактором є вимога до низької латентності системи. У VR-додатках критичним параметром є так звана затримка *motion-to-photon*, тобто час від моменту руху голови користувача до відображення відповідної зміни зображення у шоломі віртуальної реальності. Для забезпечення комфортного користування VR-системами цей показник повинен бути меншим за 20 мс. Перевищення цього значення може викликати дискомфорт або запаморочення у користувачів.

Додаткову складність створюють апаратні обмеження автономних VR-пристроїв, таких як гарнітури Oculus Quest або аналогічні системи. Такі пристрої мають обмежені обчислювальні ресурси порівняно з настільними комп'ютерами, що потребує ретельної оптимізації графіки та програмної логіки.

Архітектурні фактори пов'язані зі структурою програмної системи, що реалізує VR-додаток. У сучасних ігрових рушіях, зокрема Unreal Engine, програмна логіка може реалізовуватися як за допомогою традиційного текстового програмування мовою C++, так і за допомогою систем візуального програмування Blueprint. Співвідношення між цими підходами суттєво впливає на складність проєкту, оскільки великі графи Blueprint можуть бути складними для підтримки та аналізу.

Важливим архітектурним фактором є також складність ієрархії Actor-компонентів у рушії Unreal Engine. VR-додатки часто використовують складні композиції компонентів для реалізації інтерактивних об'єктів, систем взаємодії та механік руху.

Крім того, багато проєктів використовують стандартні шаблони, такі як VR Template в Unreal Engine 5, який містить базову реалізацію систем переміщення та взаємодії з об'єктами. Використання таких шаблонів може зменшувати складність початкової розробки, але при значній модифікації їх функціональності складність системи знову зростає.

Дизайнерські фактори пов'язані з особливостями проєктування взаємодії користувача з віртуальним середовищем. Одним із ключових аспектів є тип локомоції користувача у VR-просторі. Найбільш поширеними підходами є телепортація та плавний рух. Телепортація є простішою у реалізації та зменшує ризик появи ефекту нудоти, проте вона обмежує природність переміщення. Плавний рух забезпечує більш реалістичний досвід, але потребує складнішої реалізації та додаткових механізмів стабілізації візуального сприйняття.

Одним з дизайнерських факторів є складність систем взаємодії з об'єктами. У VR-додатках користувачі можуть виконувати різноманітні дії: хватання об'єктів, кидання, використання двох рук. Кожна з цих механік вимагає реалізації фізичної взаємодії, відстеження положення контролерів та відповідної обробки подій у програмному коді.

Проєктні фактори пов'язані з організацією процесу розробки. До них належать досвід команди у створенні VR-додатків, наявність необхідного обладнання для тестування та рівень підготовки спеціалістів. Наприклад, відсутність достатньої кількості VR-гарнітур для тестування може значно ускладнити процес перевірки функціональності системи та збільшити час розробки.

Складність проєктів розробки VR-ігор визначається сукупністю технологічних, архітектурних, дизайнерських та організаційних факторів, які необхідно враховувати при формуванні методу оцінювання складності.

У сучасних ігрових рушіях значна частина логіки гри може реалізовуватися за допомогою систем візуального програмування. У Unreal Engine такою системою є Blueprint, яка дозволяє створювати програмну логіку у вигляді графів, що складаються з вузлів та зв'язків між ними. Незважаючи на зручність використання такого підходу, зі зростанням розміру проєкту графі Blueprint можуть ставати надзвичайно складними для аналізу та підтримки.

Для кількісної оцінки складності систем візуального програмування доцільно використовувати формалізовані метрики.

Однією з базових метрик є циклометрична складність графа візуального програмування, яка визначає кількість незалежних шляхів виконання програми. Дана метрика дозволяє оцінити складність логічної структури алгоритму.

Циклометрична складність визначається за формулою:

$$CC_{BP} = E - N + 2P \quad (2.1)$$

де E – кількість ребер (зв'язків між вузлами),

N – кількість вузлів (функціональних блоків),

P – кількість незалежних графів.

Чим більше значення цієї метрики, тим складнішою є структура алгоритму та тим більше тестових сценаріїв необхідно для перевірки його коректності.

Ще однією важливою метрикою є індекс вкладеності візуального програмування, який відображає складність розуміння програмної логіки.

Ця метрика визначається за формулою:

$$NI_{BP} = \sum_{i=1}^n d_i \cdot w_i \quad (2.2)$$

де d_i – глибина вкладеності функції;

w_i – ваговий коефіцієнт типу вузла.

Використання вагових коефіцієнтів дозволяє враховувати різну складність різних типів вузлів у системі Blueprint. На основі розглянутих факторів та метрик складності можна сформуванати узагальнену послідовність оцінювання складності проєктів розробки VR-ігор.

Першим етапом є декомпозиція проєкту на основні компоненти системи віртуальної реальності. Зазвичай VR-додаток складається з кількох функціональних підсистем, зокрема системи переміщення користувача, системи взаємодії з об'єктами, користувацького інтерфейсу та дизайну рівнів. Розподіл системи на такі компоненти дозволяє виконувати оцінювання складності кожної підсистеми окремо.

Другим етапом є оцінка базової складності програмної логіки, яка визначається на основі метрик візуального та текстового програмування. Для графів Blueprint можуть використовуватися метрики циклометричної складності та індекс вкладеності, тоді як для текстового коду можуть застосовуватися традиційні метрики складності програмного забезпечення.

Третім етапом є застосування спеціалізованих коефіцієнтів складності віртуальної реальності. Такі коефіцієнти враховують специфічні особливості VR-розробки, зокрема складність реалізації систем відстеження рухів, необхідність оптимізації продуктивності та особливості взаємодії користувача з віртуальним середовищем.

Четвертим етапом є агрегація отриманих оцінок складності з урахуванням ризиків проєкту. До таких ризиків можуть належати наявність технічного боргу, необхідність оптимізації продуктивності або інтеграція з новими апаратними

платформами. Урахування цих факторів дозволяє отримати більш реалістичну оцінку складності проєкту.

Таким чином, запропонований підхід до оцінювання складності проєктів розробки VR-ігор базується на поєднанні аналізу факторів складності, використанні формалізованих метрик програмної логіки та поетапній процедурі оцінювання складності проєкту.

2.2 Розроблений метод оцінювання трудомісткості проєктів віртуальної реальності

Оцінювання трудомісткості проєктів розробки ігор віртуальної реальності є складною задачею, що потребує врахування як класичних підходів до оцінювання програмного забезпечення, так і специфічних характеристик VR-систем. У попередньому підрозділі було визначено основні фактори складності та підходи до їх формалізації. На цій основі розроблено метод оцінювання трудомісткості, який базується на адаптації параметричних моделей та введенні спеціалізованих коефіцієнтів, що враховують особливості віртуальної реальності.

Базою для оцінювання трудомісткості проєктів віртуальної реальності обрано параметричну модель СОСОМО II, яка широко використовується для прогнозування витрат на розробку програмного забезпечення. Однак у класичному вигляді ця модель не враховує особливості VR-розробки, зокрема використання систем візуального програмування та специфіку інтерактивних механік.

У зв'язку з цим запропоновано адаптовану модель, яка враховує як текстовий код, так і логіку, реалізовану у вигляді графів візуального програмування.

Адаптована модель має вигляд формули:

$$Effort_{VR} = a \cdot (KLOC_{BP} + KLOC_{C++})^b \cdot \prod_{i=1}^n E M_i \cdot VR_{factor} \quad (2.3)$$

де $KLOC_{BP}$ – еквівалентна кількість рядків коду для Blueprint (1 вузол $\approx$ 5 SLOC),

$KLOC_{C++}$ – кількість рядків коду C++;

EM_i – множники зусиль COCOMO II;

VR_{factor} – інтегральний коефіцієнт специфіки віртуальної реальності.

Для врахування внеску візуального програмування введено поняття еквівалентної кількості рядків коду. Вважається, що один вузол у графі Blueprint приблизно відповідає п'яти рядкам текстового коду, що дозволяє виконати уніфіковану оцінку обсягу програмної логіки.

Для підвищення точності оцінювання трудомісткості необхідно враховувати специфічні особливості VR-додатків, зокрема вимоги до продуктивності та складність реалізації інтерактивних механік. З цією метою введено додаткові коефіцієнти складності.

Першим із них є множник продуктивності віртуальної реальності virtual reality productivity multiplier (VPM), який враховує вплив вимог до частоти кадрів та стереоскопічного рендерингу на трудомісткість розробки. Він обчислюється за формулою:

$$VPM = 1 + 0.3 \cdot \left(1 - \frac{FPS_{target}}{FPS_{baseline}}\right) + 0.2 \cdot Stereo_{penalty}$$

де $FPS_{target} = 90$,

$FPS_{baseline}$ – досягнута продуктивність,

$Stereo_{penalty}$ – надбавка за стереорендеринг.

Дана метрика дозволяє врахувати додаткові витрати на оптимізацію продуктивності, які є критичними для VR-додатків.

Другим важливим показником є індекс складності взаємодії Virtual Reality Interaction Complexity Index (VICI), який визначає сумарну складність усіх інтерактивних механік у проєкті. Він обчислюється за формулою:

$$VICI = \sum_{j=1}^m C_j \cdot I_j$$

де C_j – складність j -ї механіки віртуальної реальності,

I_j – кількість інстансів механіки.

Наприклад, система телепортації має меншу складність порівняно з системою двуручної взаємодії, яка потребує складнішої логіки обробки рухів та фізики.

Запропонований метод оцінювання трудомісткості реалізується у вигляді послідовності етапів, кожен з яких відповідає окремому аспекту аналізу проєкту.

Етап 1. Попередній аналіз проєкту.

На цьому етапі визначаються загальні характеристики проєкту:

Крок 1.1. Визначення цільової платформи віртуальної реальності та її апаратних обмежень.

Крок 1.2. Аналіз референсних проєктів, що дозволяє отримати орієнтовні оцінки трудомісткості.

Етап 2. Декомпозиція структури робіт з компонентами віртуальної реальності.

Виконується структуризація проєкту:

Крок 2.1. Виділення базових ігрових механік.

Крок 2.2. Ідентифікація систем користувацького інтерфейсу.

Крок 2.3. Планування дизайну рівнів з урахуванням вимог VR.

Крок 2.4. Формування підходів до оптимізації продуктивності.

Етап 3. Оцінка складності систем візуального програмування.

Крок 3.1. Аналіз графів Blueprint із використанням метрик циклометричної складності (формула 2.1) та індексу вкладеності (формула 2.2).

Крок 3.2. Перетворення результатів у еквівалентну кількість рядків коду.

Етап 4. Розрахунок інтегрованої оцінки трудомісткості.

Крок 4.1. Розрахунок базової трудомісткості за адаптованою моделлю (формула 2.3).

Крок 4.2. Врахування ризиків, обчислюється за формуло:

$$Effort_{total} = Effort_{VR} \cdot (1 + Risk_{margin}),$$

де $Risk_{margin}$ – резерв на непередбачені ризики специфіки віртуальної реальності.

Етап 5. Валідація через експертну оцінку.

Крок 5.1. Порівняння з аналогічними проєктами.

Крок 5.2. Коригування параметрів моделі.

3 РОЗРОБКА ПРОЄКТУ ВПРОВАДЖЕННЯ МЕТОДУ ОЦІНЮВАННЯ СКЛАДНОСТІ В СТУДІЇ РОЗРОБКИ ІГОР

3.1 Ініціалізація та розробка концепції проєкту впровадження

У сучасних умовах розробки ігор віртуальної реальності питання точності оцінювання трудомісткості проєктів набуває критичного значення. Як було встановлено у попередніх розділах, більшість існуючих методів оцінювання не враховує специфіку VR-технологій, що призводить до систематичних відхилень між плановими та фактичними показниками проєктів.

Основною бізнес-ціллю проєкту впровадження є підвищення ефективності управління проєктами шляхом зменшення відхилень у термінах виконання та бюджетах VR-проєктів. Очікується, що застосування розробленого методу дозволить знизити рівень похибки оцінювання на 25–40%, що є критично важливим для невеликих студій розробки.

Технічною ціллю проєкту є створення інструментарію для автоматизованого оцінювання складності проєктів, який інтегрується з середовищем розробки ігрового двигуна та дозволяє аналізувати структуру візуального програмування, обчислювати відповідні метрики та формувати оцінки трудомісткості.

Організаційною ціллю є формалізація процесу планування у студії розробки ігор, що передбачає стандартизацію підходів до оцінювання складності, зменшення залежності від суб'єктивних експертних оцінок та підвищення прозорості процесів управління проєктами.

Проєкт впровадження методу оцінювання складності включає широкий спектр зацікавлених сторін, які мають різний рівень впливу на результати проєкту та різні очікування щодо його реалізації.

До внутрішніх зацікавлених сторін належать:

- проєктний менеджер, який відповідає за планування, координацію та контроль виконання проєкту;
- технічний лід, який визначає архітектурні рішення та оцінює технічну складність проєкту;
- геймдизайнер, який формує концепцію ігрових механік та впливає на рівень складності взаємодії;
- розробники, які безпосередньо реалізують програмну логіку та використовують результати оцінювання для планування задач.

До зовнішніх зацікавлених сторін належать:

- інвестори або видавці, які потребують точних оцінок для планування фінансування та контролю виконання етапів проєкту;
- партнери з тестування, які зацікавлені у стабільності процесу розробки та прогнозованості етапів тестування;
- кінцеві користувачі, які опосередковано впливають через вимоги до якості та продуктивності продукту.

Найбільший вплив на проєкт мають проєктний менеджер та технічний лід, оскільки саме вони використовують результати оцінювання для прийняття управлінських рішень.

Проєкт впровадження методу оцінювання складності реалізується в умовах обмежених ресурсів, що є типовим для невеликих студій розробки ігор.

Бюджетні обмеження – загальний бюджет проєкту становить близько 50 000 – 80 000 грн, що обмежує можливість залучення великої кількості спеціалістів або використання дорогих програмних рішень.

Часові обмеження – тривалість проєкту становить приблизно 3 місяці, що передбачає необхідність чіткої організації робіт та ефективного використання часу.

Ресурсні обмеження – до виконання проєкту залучається обмежена команда з проєктного менеджера, старшого розробника та допоміжних технічних ресурсів.

Інформаційні обмеження – доступ до історичних даних про попередні проекти обмежений, що може впливати на точність калібрування методу.

Таким чином, проект впровадження повинен бути спроектований з урахуванням принципів мінімізації витрат та максимальної ефективності використання доступних ресурсів.

3.2 Розробка життєвого циклу проекту впровадження

З урахуванням специфіки проекту впровадження доцільним є використання гібридного підходу до управління, що поєднує елементи гнучких та класичних методологій.

Agile-підхід використовується для розробки програмного інструментарію, оскільки дозволяє реалізовувати функціональність ітеративно, забезпечує швидкий зворотний зв'язок, та дає можливість адаптувати рішення під потреби команди.

Waterfall-підхід застосовується для організаційних змін, таких як впровадження нового процесу оцінювання, навчання команди, формалізація документації.

Такий комбінований підхід дозволяє забезпечити гнучкість розробки інструментарію та водночас зберегти структурованість впровадження організаційних змін.

Життєвий цикл проекту включає такі основні фази:

- ініціація;
- планування;
- виконання;
- моніторинг і контроль;
- закриття.

На ініціацію виділяється 2 тижні та на ньому здійснюється визначення цілей проєкту, збір вимог, аналіз наявних даних, формування концепції рішення.

На планування виділяється 3 тижні та цей етап включає розробку структури робіт Work Breakdown Structure (WBS), оцінювання ресурсів, формування календарного плану, визначення ризиків.

На виконання виділяється 6 тижнів та на ньому реалізуються розробка програмного інструментарію, інтеграція з системами управління проєктами, пілотне застосування.

Моніторинг і контроль виконується паралельно з виконанням та передбачає контроль виконання задач, відстеження метрик, аналіз відхилень, коригування плану.

На закриття виділяється 1 тиждень та він включає ретроспективу, оцінку результатів, документування досвіду, передачу інструменту у використання.

3.3 Структуризація проєкту впровадження

Для забезпечення ефективного планування та контролю виконання проєкту впровадження методу оцінювання складності доцільно застосувати структурну декомпозицію робіт. Декомпозиція дозволяє представити проєкт у вигляді ієрархічної структури взаємопов'язаних робіт, що забезпечує прозорість процесу управління та спрощує розподіл відповідальності між учасниками команди.

У межах даного проєкту WBS включає п'ять основних етапів:

- аналітичну фазу;
- проєктування рішення;
- розробку інструментарію;
- впровадження та навчання;
- оцінка ефективності.

Метою аналітичної фази є формування інформаційної бази для розробки методу оцінювання та визначення вимог до майбутньої системи, та включає в себе збір історичних даних метрик візуального програмування та фактичних витрат, інтерв'ю з командою щодо проблем планування, benchmark існуючих інструментів.

На етапі проєктування рішення визначається архітектура системи автоматизації, інтеграційні точки з редактором двигуна та Application Programming Interface (API) систем управління проєктами, створюються шаблони звітності.

Етап розробка інструментарію передбачає створення програмної реалізації методу оцінювання, а саме розробка скриптів парсингу Blueprint, реалізація модуля розрахунку складності VR-коефіцієнтів, розробка системи візуалізації.

Метою етапу навчання та впровадження є інтеграція розробленого методу у реальні процеси студії, на ньому проводяться навчання команди, пілотне впровадження та розробка документації.

Завершальний етап оцінки ефективності спрямований на аналіз результативності проєкту, включає в себе порівняльний аналіз результатів, оцінку задоволеності користувачів, економічну оцінку ефективності.

Ефективна реалізація проєкту впровадження методу оцінювання складності потребує чіткого визначення та структуризації ресурсів. У межах даного проєкту ресурси доцільно класифікувати за чотирма основними групами: управлінські, технічні, консультаційні та матеріальні.

Ієрархічна структура ресурсів відображає їх роль у проєкті, рівень відповідальності та ступінь залучення до виконання робіт.

Управлінські ресурси забезпечують організацію, координацію та контроль виконання проєкту.

Управлінським ресурсом в проєкті є Project Manager (PM). Він виконує роль керівника проєкту, рівень його залучення дорівнює 0.5 Full-Time Equivalent (FTE).

Основні функції полягають у планування проєкту, координація роботи команди, контроль виконання задач і дотримання строків, управління ризиками, комунікація зі стейкхолдерами.

У невеликих інді-студіях роль РМ часто поєднується з іншими функціями, що обумовлює часткову зайнятість у проєкті.

Технічні ресурси є ключовими для реалізації програмної частини системи оцінювання.

Першим технічним ресурсом є Senior Gameplay Programmer. Його роль – основний виконавець технічної реалізації, рівень залучення дорівнює 0.3 FTE.

Ключові компетенції полягають у роботі з Unreal Engine 5, розробці Blueprint-логіки, програмуванні на Python.

Основні задачі включають розробку скриптів аналізу Blueprint-графів, реалізацію алгоритмів розрахунку складності, інтеграцію модулів оцінювання, тестування коректності роботи системи.

Другим технічним ресурсом є DevOps Engineer. Він виконує роль підтримки інфраструктури та інтеграцій, рівень залучення дорівнює 0.1 FTE.

Основні функції складаються з налаштування Continuous Integration/Continuous Deployment (CI/CD) процесів, інтеграція інструменту з системами управління проєктами, забезпечення стабільності роботи сервісів, автоматизація розгортання системи.

Залучення DevOps-інженера є обмеженим, оскільки проєкт не передбачає створення складної інфраструктури, проте його участь є критичною для забезпечення інтеграції та автоматизації процесів.

Консультаційні ресурси використовуються для підвищення якості методологічної частини проєкту.

Консультаційним ресурсом проєкту є зовнішній експерт з оцінювання проєктів. Його рівень залучення є епізодичним, на 2 консультації.

Основні задачі полягають у перевірці коректності застосування методів оцінювання, аналізі адекватності коефіцієнтів складності, наданні рекомендацій щодо вдосконалення моделі, участь у валідації результатів.

Залучення зовнішнього експерта дозволяє зменшити ризик методологічних помилок та підвищити обґрунтованість отриманих результатів.

Матеріальні ресурси забезпечують технічну можливість реалізації та тестування системи.

Матеріальні ресурси включають програмне забезпечення та інструменти: середовище розробки Unreal Engine 5, мова програмування Python, системи управління проєктами Jira, інструменти візуалізації Grafana або Tableau.

Також потрібні тестові дані та проєкти: один або декілька VR-проєктів для пілотного впровадження, історичні дані завершених проєктів, файли Blueprint для аналізу.

Опціонально також можна включати додаткові ресурси: ліцензії на аналітичні або інструменти Business Intelligence (BI), хмарна інфраструктура для розміщення dashboard, VR-обладнання для тестування.

3.4 Проєктування системи оцінювання складності проєктів віртуальної реальності

Для реалізації методу оцінювання складності проєктів розробки ігор віртуальної реальності пропонується багаторівнева архітектура програмного рішення, яка забезпечує збір, обробку, інтеграцію та візуалізацію даних. Архітектура побудована за принципом розділення відповідальностей і включає чотири основні рівні: Input layer, Processing layer, Integration layer та Output layer.

Input layer відповідає за отримання первинних даних про структуру проєкту з ігрового рушія. Основним компонентом цього рівня є плагін редактора ігрового двигуна. Він реалізується як розширення для Unreal Editor, використовує Reflection System Unreal Engine для доступу до метаданих, забезпечує автоматичне зчитування інформації про Blueprint-графи.

Функції цього рівня полягають у отримання списку Blueprint-об'єктів проєкту, аналіз структури графів, збір метрик, кількість вузлів, кількість зв'язків, кількість умовних переходів, кількість незалежних графів, формування структурованого представлення даних. Результат роботи Input layer це набір сирих метрик складності, що передається до рівня обробки. Processing layer є ядром системи та відповідає за обчислення показників складності та трудомісткості. Основним компонентом є Backend-сервіс на Python.

Функціями рівня Processing є обчислення метрик складності, реалізація моделі оцінювання, агрегація результатів, використання моделей машинного навчання. Результат роботи цього рівня розраховані показники складності та трудомісткості проєкту.

Integration layer забезпечує взаємодію системи оцінювання з зовнішніми інструментами управління проєктами. Основним компонентом є REST API.

Функціями рівня Integration є передача результатів оцінювання у системи управління проєктами, синхронізація даних між компонентами системи, забезпечення доступу до результатів через API, інтеграція з зовнішніми системами Jira та Azure DevOps.

Output layer відповідає за візуалізацію результатів оцінювання та їх інтерпретацію користувачами. Основним компонентом є Web-dashboard.

Функціями рівня Output є відображення інтегральної оцінки складності, візуалізація структури проєкту, побудова графіків і діаграм, аналіз динаміки складності. Ключовими елементами інтерфейсу є теплова карта складності

компонентів, графіки залежності складності від часу, порівняння планових та фактичних показників, відображення ризиків.

Взаємодія між рівнями архітектури відбувається у такій послідовності: Input layer отримує дані з ігрового рушія → Processing layer виконує обробку та розрахунки → Integration layer передає результати у системи управління → Output layer відображає результати у зручному вигляді.

Логіко-інформаційна схема процесу оцінювання складності проєктів віртуальної реальності визначає послідовність обробки даних, взаємодію між компонентами системи та розподіл відповідальності між учасниками процесу.

Основною метою побудови такої схеми є формалізація процесу оцінювання, забезпечення прозорості розрахунків та можливість інтеграції результатів у систему управління проєктами.

Процес оцінювання включає декілька взаємопов'язаних етапів: аналіз структури візуального програмування проєкту, розрахунок базової трудомісткості, застосування коефіцієнтів віртуальної реальності, формування структури робіт із оцінками та моніторинг відхилень від плану.

Задачі, вихідна інформація, виконавці та споживачі приведені в таблиці 3.1.

Таблиця 3.1 – Процеси системи оцінювання

Задача	Вихідна інформація	Споживач результату
Аналіз структури візуального програмування проєкту	Файли проєкту, концепт-документ	Менеджер проєкту, технічний лід
Розрахунок базової трудомісткості	Метрики складності, історичні дані	Менеджер проєкту
Застосування коефіцієнтів віртуальної реальності	Цільова платформа, тип взаємодій	Менеджер проєкту
Формування структури робіт із оцінками	Агреговані оцінки, залежності завдань	Команда проєкту
Моніторинг відхилень від плану	Фактичні витрати часу, базовий план	Менеджер проєкту, стейкхолдери

Логіко-інформаційна схема процесу оцінювання забезпечує формалізований підхід до аналізу складності VR-проектів, дозволяє автоматизувати ключові етапи оцінювання та інтегрувати результати у процес управління проектами. Це створює основу для підвищення точності планування та зменшення ризиків відхилень у термінах і бюджетах.

Для планування та контролю виконання проекту впровадження методу оцінювання складності доцільно використовувати мережеве моделювання. Побудова мережевого графіка дозволяє визначити послідовність виконання робіт, встановити залежності між ними, а також виявити критичний шлях проекту. Перелік робіт та їх залежності було занесено у таблицю 3.2, а сам мережевий графік зображено на рисунку 3.1.

Таблиця 3.2 – Роботи мережевого графіка

№	Назва задачі	Тривалість	Попередні задачі	Відповідальний
1	Ініціація проекту	2	-	Project Manager
2	Збір історичних даних	5	1	Developer
3	Аналіз процесів планування	4	1	Project Manager
4	Формування вимог до системи	3	2,3	Project Manager
5	Проектування архітектури	4	4	Developer
6	Розробка парсера Blueprint	10	5	Developer
7	Реалізація модуля розрахунків	6	5	Developer
8	Інтеграція з системами управління проектами	5	6,7	DevOps, Developer
9	Розробка dashboard	5	7	Developer
10	Пілотне застосування	5	8,9	Вся команда
11	Аналіз результатів	3	10	Project Manager
12	Валідація точності оцінювання	3	11	Project Manager
13	Документування та завершення	2	12	Project Manager

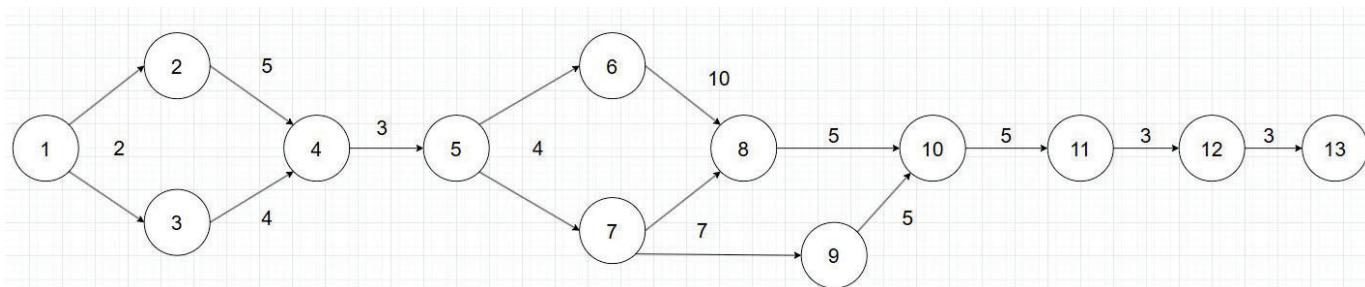


Рисунок 3.1 – Мережевий графік

Критичний шлях визначає мінімально можливу тривалість проєкту та включає роботи, затримка яких безпосередньо призводить до зміщення строків завершення всього проєкту.

У межах даного проєкту критичний шлях має вигляд:

$$1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 8 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 13.$$

Сумарна тривалість критичного шляху становить:

$$2 + 5 + 3 + 4 + 10 + 5 + 5 + 3 + 3 + 2 = 42 \text{ дні.}$$

3.5 Керування часом, вартістю та ресурсами проєкту

Календарний план проєкту реалізовано у вигляді діаграми Ганта, яка відображає послідовність виконання робіт, їх тривалість та взаємозв'язки. План передбачає поділ проєкту на три основні фази: аналітику та проєктування, розробку інструментарію та впровадження з подальшою оцінкою результатів. Такий підхід дозволяє забезпечити паралельне виконання частини задач, оптимізувати використання ресурсів та скоротити загальну тривалість проєкту. Діаграма Ганта зображена на рисунку 3.2.

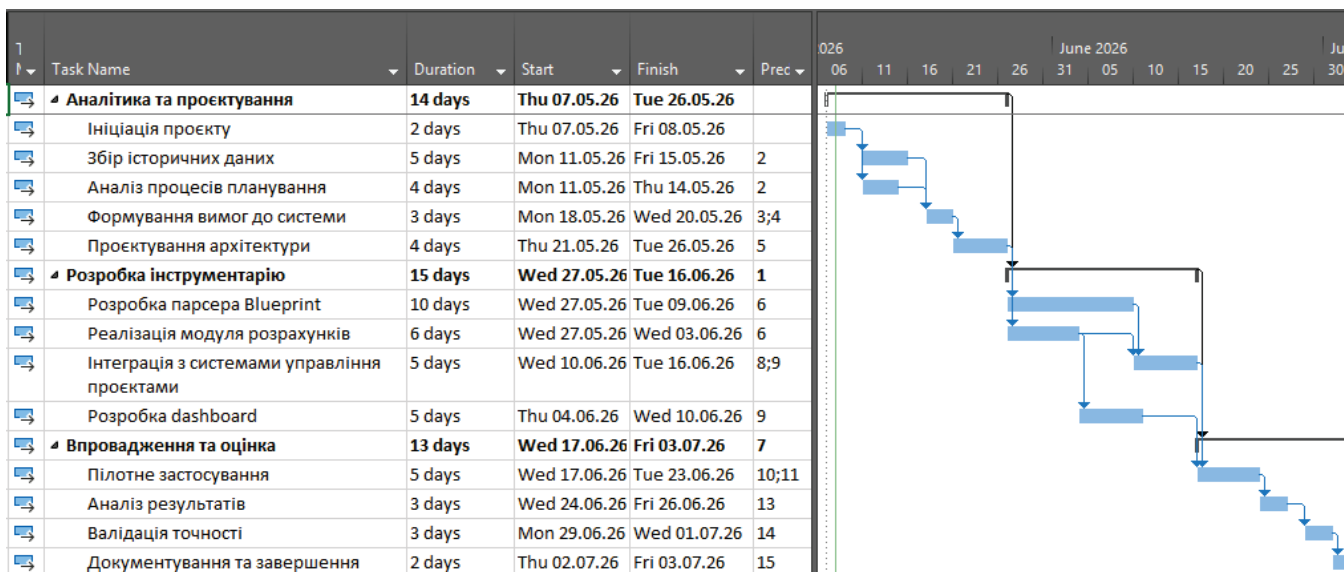


Рисунок 3.2 – Діаграма Ганта

Бюджет проєкту впровадження методу оцінювання складності сформовано з урахуванням обмежених ресурсів інді-студії та передбачає оптимальне використання наявного персоналу. Основну частину витрат становлять витрати на оплату праці, що є характерним для ІТ-проєктів. Основні витрати проєкту буде приведено нижче у таблиці 3.3.

Таблиця 3.3 – Витрати проєкту

Стаття витрат	Вартість (грн)	Обґрунтування
Менеджер проєкту (0.5 FTE * 3 міс)	30 000	Координація проєкту, планування, контроль виконання, документування
Старший розробник (0.3 FTE * 3 міс)	27 000	Розробка інструментарію, реалізація алгоритмів оцінювання
Зовнішні консультації	8 000	2 * 4 години експертної оцінки
Програмне забезпечення та інструменти	3 000	Плагіни, хмарне розміщення dashboard
Навчальні матеріали	2 000	Створення документації, воркшопи
Резерв (15%)	10 500	Непередбачені витрати
ЗАГАЛОМ	80 500	

Сформований бюджет проєкту є реалістичним для умов невеликої студії розробки та забезпечує необхідний рівень ресурсів для реалізації проєкту впровадження. Його структура відповідає специфіці ІТ-проєктів, де ключовим фактором витрат є людські ресурси.

Управління ризиками є важливою складовою процесу реалізації проєкту впровадження методу оцінювання складності, оскільки проєкт виконується в умовах обмежених ресурсів, невизначеності та використання нових підходів. Основною метою управління ризиками є своєчасне виявлення потенційних загроз, оцінка їх впливу на проєкт та розробка заходів реагування для мінімізації негативних наслідків.

Перелік ризиків проєкту буде наведений нижче у таблиці 3.4.

Таблиця 3.4 – Ризики проєкту

№	Ризик	Ймовірність	Вплив	Рівень ризику	Стратегія реагування
1	Недостатня точність оцінок через обмежені історичні дані	Середня	Високий	Високий	Калібрування коефіцієнтів на основі пілотного проєкту, залучення експертних оцінок
2	Опір команди впровадженню нового підходу	Середня	Середній	Середній	Залучення team leads до розробки рішення, демонстрація результатів
3	Технічні складнощі при парсингу Blueprint	Середня	Високий	Високий	Використання Python API Unreal Engine, резервний варіант введення метрик
4	Недостатня інтеграція з системами управління проєктами	Низька	Середній	Низький	Використання стандартних REST API, тестування інтеграцій на ранніх етапах
5	Перевищення запланованого бюджету	Низька	Середній	Низький	Використання резервного фонду (15%), контроль витрат
6	Затримки у виконанні критичних задач	Середня	Високий	Високий	Контроль критичного шляху, регулярний моніторинг виконання задач

Запропонована система управління ризиками дозволяє своєчасно ідентифікувати потенційні загрози та забезпечує механізми їх мінімізації. Це підвищує ймовірність успішної реалізації проєкту впровадження та досягнення поставлених цілей у визначені строки та бюджет.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА

4.1 Реалізація системи автоматизованого оцінювання складності проєктів

Для практичної реалізації запропонованого методу оцінювання складності проєктів віртуальної реальності розроблено програмний модуль, архітектура якого побудована за модульним принципом і включає чотири основні компоненти: плагін редактора ігрового рушія, обчислювальний модуль, API-шар інтеграції та підсистему візуалізації.

Така архітектура забезпечує розподіл функціональності, масштабованість рішення та можливість інтеграції з існуючими інструментами розробки і управління проєктами.

Першим рівнем системи є плагін для редактора Unreal Engine 5, реалізований мовою C++. Його основним призначенням є автоматизований збір метаданих візуального програмування.

Основні функції плагіна є інтеграція з Blueprint Compiler для доступу до структури графів, отримання інформації про кількість вузлів, кількість зв'язків, кількість незалежних шляхів виконання, аналіз структури Blueprint-класів та їх залежностей, формування структурованого представлення даних.

Плагін використовує внутрішні механізми рушія, що забезпечує високу точність отриманих метрик та мінімальний вплив на продуктивність редактора.

Результатом роботи є набір метрик складності у вигляді структурованих даних, що передаються до обчислювального модуля.

Центральним компонентом системи є обчислювальний модуль, реалізований мовою Python, який виконує всі розрахунки, пов'язані з оцінюванням складності.

Основні функції модуля складаються з розрахунків метрик складності, реалізація моделі трудомісткості, застосування VR-коефіцієнтів, робота з історичними даними.

Розрахунок метрик складності складається з обчислення циклометричної складності Blueprint-графів, розрахунок індексу вкладеності, визначення еквівалентного обсягу коду.

Реалізація моделі трудомісткості передбачає застосування базової параметричної моделі, обчислення інтегральної оцінки складності, агрегація результатів для всього проєкту.

Для застосування VR-коефіцієнтів необхідно врахування продуктивності, врахування складності взаємодії, корекція базової оцінки відповідно до специфіки VR.

Робота з історичними даними складається з збереження результатів попередніх оцінок, корекція коефіцієнтів, підвищення точності прогнозування.

Результатом роботи є інтегральна оцінка трудомісткості та складності проєкту.

Для забезпечення взаємодії між компонентами системи та зовнішніми сервісами використовується API-шар, реалізований на базі Flask.

Основні функції API:

- надання REST endpoints для доступу до результатів оцінювання;
- передача даних між плагіном, обчислювальним модулем і frontend;
- інтеграція з системами управління проєктами.

API-шар передбачає такі основні можливості, як отримання метрик складності, передача результатів у зовнішні системи Jira, Azure DevOps, підтримка стандартних форматів обміну даними JavaScript Object Notation (JSON). Він забезпечує слабе зв'язування компонентів системи, що дозволяє легко масштабувати та модифікувати рішення.

Візуалізація результатів оцінювання реалізована у вигляді web-додатку на базі React. Функції dashboard складаються з відображення показників складності проєкту, візуалізація структури компонентів, порівняння оцінок із фактичними витратами, аналіз відхилень.

Інтерфейс додатку складається з графіків складності, теплових карт, таблиць оцінок, динаміка змін метрик у часі.

Використання сучасних бібліотек візуалізації забезпечує наочність та зручність інтерпретації результатів.

Робота системи відбувається за такою схемою: UE5 Plugin збирає метрики з Blueprint → дані передаються до Python Calculation Engine → виконується розрахунок складності та трудомісткості → результати передаються через Web API → dashboard відображає результати користувачу → за потреби дані інтегруються в систему управління проєктами.

У межах розробленого програмного модуля системи оцінювання складності ключову роль відіграють алгоритми аналізу структури візуального програмування та обчислення метрик складності. Основна увага приділяється автоматизованому парсингу Blueprint-графів та розрахунку їх характеристик.

Алгоритм парсингу реалізовано з використанням Python API ігрового рушія Unreal Engine. Його метою є отримання структурних характеристик Blueprint-графа, необхідних для подальших обчислень.

Основні етапи алгоритму:

- завантаження Blueprint asset із проєкту;
- отримання доступу до графа виконання;
- ітерація по вузлах графа;
- визначення зв'язків між вузлами;
- підрахунок кількості вузлів, кількості зв'язків, кількості компонент зв'язності;
- формування структури даних для подальшого аналізу.

Ключовим показником є циклометрична складність Blueprint-графа, яка визначається за формулою 2.1. Він дозволяє оцінити кількість незалежних шляхів виконання програми та складність її логіки.

На основі отриманих метрик реалізується алгоритм розрахунку складності:

- отримання значень N, E, P;
- обчислення циклометричної складності;
- розрахунок додаткових метрик (наприклад, індексу вкладеності);
- агрегація результатів;
- передача даних у модуль візуалізації або API.

Лістинг функції розрахунку складності Blueprint-графа, як і методу парсингу Blueprint наведено у додатку А.

Розроблені алгоритми інтегруються у загальну архітектуру таким чином: UE5 Plugin ініціює збір даних → Python-модуль виконує обчислення → результати передаються через API → frontend відображає отримані показники.

Запропоновані алгоритми забезпечують автоматизований аналіз структури візуального програмування та дозволяють кількісно оцінити складність Blueprint-графів. Їх програмна реалізація на Python є гнучкою, розширюваною та може бути інтегрована у реальні процеси розробки VR-проектів.

Для забезпечення практичного використання результатів оцінювання складності у процесі планування проектів реалізовано інтеграцію програмного модуля із системами управління проектами. Основним інструментом інтеграції обрано Jira, що широко використовується в IT-проектах.

Інтеграція реалізується через REST API та дозволяє автоматизувати створення задач, передачу оцінок складності та формування структури робіт на основі результатів аналізу.

Взаємодія з системою управління проектами здійснюється через Jira REST API, що забезпечує можливість програмного створення та модифікації задач.

Основні можливості інтеграції:

- створення епиків;
- створення user stories;
- додавання кастомних полів;

- оновлення оцінок;
- отримання даних про виконання задач.

Формат обміну даними: JSON

На основі результатів оцінювання система автоматично створює епіки, які представляють великі функціональні блоки проєкту.

До кожного епіка додаються кастомні поля Estimated Complexity та VR Risk Factor. Estimated Complexity розраховується за формалізованою моделлю та відображає інтегральну складність задачі. VR Risk Factor визначається на основі VR-коефіцієнтів та враховує специфіку взаємодії та продуктивності у VR.

Ці поля дозволяють менеджеру проєкту приймати обґрунтовані рішення щодо пріоритезації та планування.

Однією з ключових функцій є автоматичне перетворення WBS-декомпозиції у структуру задач у Jira.

Алгоритм інтеграції починається з отримання структури WBS із системи оцінювання, потім створюються Епіс для кожного верхнього рівня WBS, після цього відбувається автоматичне створення user stories для підзадач. Далі встановлюються ієрархічні зв'язки за схемою Епіс → Story → Sub-task. Далі API призначає оцінки складності кожній задачі та встановлює залежностей між задачами.

Лістинг REST-запиту створення епіка наведено у додатку А.

Інтеграція системи оцінювання складності з Jira через REST API дозволяє безпосередньо використовувати результати аналізу у процесі управління проєктами. Автоматизація створення задач та передача оцінок забезпечують узгодженість між технічним аналізом і управлінськими рішеннями, що сприяє підвищенню ефективності реалізації VR-проєктів.

4.2 Експериментальна перевірка методу на реальних проєктах

Для перевірки ефективності запропонованого методу оцінювання складності було проведено експеримент на основі реальних та пілотного проєктів розробки VR-додатків. Вибірка включає два завершених проєкти, що використовуються для калібрування та валідації методу, а також один пілотний проєкт, для якого застосовано новий підхід оцінювання.

Метою експерименту є порівняння фактичних витрат трудомісткості з оцінками, отриманими за допомогою розробленого методу, а також оцінка його придатності для прогнозування ресурсів у майбутніх проєктах.

Проєкт А – завершений. Представляє собою VR-гру жанру escape room, орієнтовану на автономні VR-пристрої. Проєкт характеризується середнім рівнем складності логіки та обмеженнями продуктивності платформи.

- тривалість розробки: 4 місяці;
- команда: 3 розробники;
- платформа: Meta Quest 2;
- фактичні витрати: 1 920 людино-годин;
- метрики Blueprint: вузлів 487, циклометрична складність 52.

Проєкт Б – завершений. Прототип VR-шутера з більш складною логікою взаємодії та розширеними можливостями обробки подій. Використання Personal Computer Virtual Reality (PCVR) платформи дозволило реалізувати більш складні механіки.

- тривалість розробки: 6 місяців;
- команда: 5 розробників;
- платформа: PCVR;
- фактичні витрати: 4 800 людино-годин;
- метрики Blueprint: вузлів 1 243, циклометрична складність 138.

Проект В – пілотний. Новий проект пригодницької VR-гри, для якого застосовано розроблений метод оцінювання складності на етапі планування.

- планова тривалість: 8 місяців;
- команда: 4 розробники;
- платформа: Meta Quest 3;
- оцінена трудомісткість: 3 200 людино-годин за новим методом.

Проект використовується для перевірки практичної застосовності методу та оцінки його точності у прогнозуванні.

Для оцінки ефективності запропонованого методу було проведено порівняння його результатів із базовим підходом оцінювання на основі завершених проєктів. Основним критерієм оцінки виступає відхилення розрахованої трудомісткості від фактичних витрат. Результат порівняння було занесено в таблицю 4.1.

Таблиця 4.1 – Таблиця точності оцінювання

Проект	Фактичні витрати (год)	Базова оцінка (PERT, год)	Розроблений метод (год)	Відхилення базової оцінки	Відхилення розробленого методу
А	1 920	1 450 (-24,5%)	1 870 (-2,6%)	-470	-50
Б	4 800	3 600 (-25%)	4 650 (-3,1%)	-1 200	-150
В	-	2 400	3 200	–	–

Результати експерименту демонструють суттєву перевагу розробленого методу над традиційним підходом. Для проєкту А похибка зменшилась з 24,5% до 2,6%, для проєкту Б — з 25% до 3,1%, абсолютне відхилення зменшилось більш ніж у 8 разів.

Це свідчить про те, що врахування структурних метрик Blueprint дозволяє значно точніше оцінювати трудомісткість VR-проєктів.

Для кількісної оцінки точності розробленого методу було використано стандартні статистичні показники, зокрема середню абсолютну відносну похибку та коефіцієнт кореляції Пірсона.

Середня абсолютна відносна похибка визначається за формулою:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{A_i - F_i}{A_i} \right| * 100\%$$

де A_i – фактичні значення трудомісткості;

F_i – прогнозовані значення;

n – кількість спостережень.

Отримані результати:

– базовий метод: 24,75%;

– розроблений метод: 2,85%.

Покращення становить 21,9%. Це означає, що похибка прогнозування зменшилась майже у 9 разів, що є суттєвим підвищенням точності.

Для оцінки узгодженості прогнозованих і фактичних значень було використано коефіцієнт кореляції Пірсона. Результат $r=0,94$. Це значення свідчить про сильний позитивний лінійний зв'язок між оціненими та фактичними витратами трудомісткості.

Отримані статистичні показники дозволяють зробити такі висновки:

– значення Mean Absolute Percentage Error (MAPE) < 10% для розробленого методу відповідає високій точності прогнозування;

– коефіцієнт кореляції $r = 0,94$ підтверджує адекватність моделі;

– базовий метод демонструє значно гірші результати через суб'єктивність оцінок, відсутність урахування структурної складності, ігнорування специфіки VR.

4.3 Оцінка впливу впровадження на процеси організації

Для оцінки впливу впровадження розробленого методу оцінювання складності на процеси організації було використано набір ключових показників ефективності.

До впровадження час на перепланування складав 12 годин на місяць, після впровадження стало 3 години на місяць. Витрати часу скорочено на 75%. Це досягнуто завдяки підвищенню точності первинного планування та зменшенню кількості коригувань.

До впровадження показника своєчасного виконання складав 40%, після впровадження стало 75%. Показник зріс на 35%. Покращення пояснюється більш точним визначенням трудомісткості та кращою синхронізацією плану з реальними можливостями команди.

Економічний ефект впровадження оцінюється через скорочення витрат часу менеджера проєкту. Економія часу складає 120 годин за 6 місяців, при середній вартості 1 години менеджера проєкту ≈ 500 грн, загальна економія буде дорівнювати $120 * 500 = 60\,000$ грн. Витрати на впровадження 80 500 грн. Щомісячна економія дорівнює $60\,000 / 6 = 10\,000$ грн/місяць. При цих показниках строк окупності буде складати $80\,500 / 10\,000 \approx 8$ місяців

Отримані метрики свідчать про значний позитивний вплив впровадження методу оцінювання складності на процеси управління проєктами. Зокрема, досягнуто істотного скорочення витрат часу на перепланування, підвищення дотримання строків виконання та забезпечено економічну доцільність впровадження.

Для оцінки зручності використання розробленого інструменту було проведено опитування членів команди розробки після його впровадження у робочий процес. Оцінювання здійснювалось із використанням стандартизованої методики System Usability Scale, яка дозволяє кількісно визначити рівень зручності програмного продукту.

За результатами опитування було отримано SUS score – 78,5

Відповідно до шкали інтерпретації оцінка відповідає рівню Grade B, якісна характеристика – “Good”.

Це свідчить про високий рівень зручності використання системи та її готовність до практичного застосування без суттєвих доопрацювань.

У межах опитування також було оцінено окремі аспекти системи за п'ятибальною шкалою:

- легкість інтерпретації метрик 4,2/5
- корисність для планування спринтів 4,5/5
- інтеграція в робочий процес 4,0/5
- зручність інтерфейсу 3,9/5
- швидкість обчислення 3,1/5

Сильні сторони системи:

- висока зрозумілість метрик складності, що полегшує прийняття рішень;
- значна корисність для планування спринтів, що підтверджує інтеграцію інструменту у процеси Agile;
- загальна позитивна оцінка досвіду користувача.

Зони для покращення:

- швидкість обчислення для великих Blueprint-графів;
- потреба в оптимізації алгоритмів або використанні кешування;
- можливість асинхронної обробки даних.

Значення 78,5 перевищує середній рівень у 68 баллів, що означає система є зручною для більшості користувачів, не потребує значного навчання та має високий потенціал для масштабування в межах організації.

Результати опитування підтверджують, що розроблений інструмент має високий рівень зручності використання та позитивно сприймається командою. Виявлені недоліки носять технічний характер і можуть бути усунені шляхом оптимізації продуктивності, що не знижує загальної ефективності рішення

ВИСНОВОК

У кваліфікаційній роботі вирішено науково-практичну задачу оцінювання складності проєктів розробки VR-ігор шляхом створення удосконаленого методу, що інтегрує метрики систем візуального програмування з коефіцієнтами специфіки віртуальної реальності.

У процесі дослідження проведено аналіз предметної області, який показав, що існуючі методи оцінювання не повною мірою враховують особливості VR-проєктів, зокрема використання візуального програмування, підвищені вимоги до продуктивності та складність інтерактивної взаємодії.

На основі аналізу розроблено метод оцінювання трудомісткості, що поєднує параметричну модель СОСОМО II з метриками складності Blueprint-графів і VR-коефіцієнтами VPM та VICI.

Реалізовано програмний модуль для автоматизованого збору метрик, обчислення складності та інтеграції результатів у системи управління проєктами. Експериментальна перевірка методу на реальних проєктах показала підвищення точності оцінювання: середня абсолютна відносна похибка MAPE зменшилась з 24,75% до 2,85%, а коефіцієнт кореляції між прогнозованими та фактичними значеннями склав 0,94, що підтверджує адекватність моделі.

Оцінка впливу впровадження методу показала покращення ключових показників управління проєктами, зокрема скорочення часу на перепланування на 75% та підвищення рівня виконання задач у встановлені строки з 40% до 75%.

Отже, розроблений метод є ефективним інструментом оцінювання складності VR-проєктів. Результати роботи можуть використовуватись менеджерами проєктів і технічними лідами студій розробки для підвищення точності планування VR-проєктів, а також як основа для подальших досліджень прогнозування трудомісткості на основі патернів візуального програмування.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлювання. – Чинний від 22.06.2015. – Київ: ДП «УкрНДНЦ», 2016. 31 с.
2. ДСТУ 8302:2015. Інформація та документація. Бібліографічні посилання. Загальні положення та правила складання. – Чинний від 04.03.2016. – Київ: ДП «УкрНДНЦ», 2016. 20 с.
3. Epic Games. Unreal Engine Documentation. URL: <https://dev.epicgames.com/documentation/> (дата звернення: 07.04.2026)
4. Rogers S. Level Up! The Guide to Great Video Game Design. – New York: McGraw-Hill Education, 2010. – 467 p.
5. Jerald J. The VR Book: Human-Centered Design for Virtual Reality. – Morgan & Claypool, 2016. – 120 p.
6. Boehm B. COCOMO II Model Definition Manual. – Upper Saddle River: Prentice Hall PTR, 2000. – 506 p.
7. Garmus D. Certified Function Point Specialist Examination Guide. – Auerbach Publications, 2010 – 154 p.
8. Boehm B. W. Software Engineering Economics. – Englewood Cliffs: Prentice Hall, 1981. – 767 p.
9. Jerald J. The VR Book: Human-Centered Design for Virtual Reality. – New York: Morgan & Claypool, 2015. – 637 p.
10. Sherman W., Craig A. Understanding Virtual Reality: Interface, Application, and Design. – Burlington: Morgan Kaufmann, 2018. – 938 p.
11. Schwaber K., Sutherland J. The Scrum Guide. – Scrum.org, 2020. – 14 p.
12. LaValle S. Virtual Reality. – Cambridge: Cambridge University Press, 2017. – 408 p.

13. Sommerville I. Software Engineering. – Boston: Pearson, 2016. – 816 p.
14. Martin R. C. Clean Code: A Handbook of Agile Software Craftsmanship. – Boston: Prentice Hall, 2008. – 464 p.
15. Sauro J., Lewis J. R. Quantifying the User Experience: Practical Statistics for User Research. – Cambridge: Morgan Kaufmann, 2016. – 352 p.
16. Blueprint Visual Scripting Documentation. URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/blueprints-visual-scripting-in-unreal-engine> (дата звернення: 10.04.2026)
17. Unreal Engine Python API Documentation. URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/python-api> (дата звернення: 12.04.2026)
18. Jira REST API Documentation. URL: <https://developer.atlassian.com/cloud/jira/platform/rest/v3/intro> (дата звернення: 20.04.2026)
19. React Documentation. URL: <https://react.dev> (дата звернення: 22.04.2026)
20. Python Documentation. URL: <https://docs.python.org/3> (дата звернення: 25.04.2026)
21. Meta Quest Developer Documentation. URL: <https://developer.oculus.com/documentation> (дата звернення: 30.04.2026)
22. Чалий С. Ф., Бганцев М. В. Розробка методів оцінювання ІТ-компаній з використанням анонімних відгуків. The 8 th International scientific and practical conference “Innovative development of science, technology and education” Perfect Publishing. Vancouver, Canada, 2024. p. 293-295.
23. Чалий С. Ф., Лещинський В. О. Метод можливісного оцінювання пояснення в системі штучного інтелекту. Bulletin of National Technical University" KhPI". Series: System Analysis, Control and Information Technologies. Kharkiv, 2023. p. 95-101.
24. Чалий С. Ф., Левикін І. В. Логічна модель реалізації бізнес-процесів. Нові технології. Харків, 2007. № 4 (18). С. 55-59.

25. Чалий С. Ф., Левикін В. М. Метод адаптивного процесного управління на основі прецедентного підходу. Наукоємні технології. Харків, 2016. № 4. С. 410-414.

26. Чалий С. Ф., Левикін І.В., Кальницька А. Ю. Постановка задачі прецедентного управління наскрізними бізнес-процесами. Автоматизовані системи управління і прилади автоматики. Харків, 2019. № 175. С. 19-26.

27. Гужва В. М. Інформаційні системи і технології на підприємствах. – Київ: КНЕУ, 2018. – 400 с.

28. Ліпаєв В. В. Оцінювання якості та складності програмних засобів. – Київ: Академперіодика, 2011. – 312 с.

29. Мельник Г. М. Аналіз сучасних методів оцінювання складності програмного забезпечення. – 2021. – №3. – С. 67–74.

30. Ситник В. Ф. Системи підтримки прийняття рішень. – Київ: КНЕУ, 2009. – 614 с.